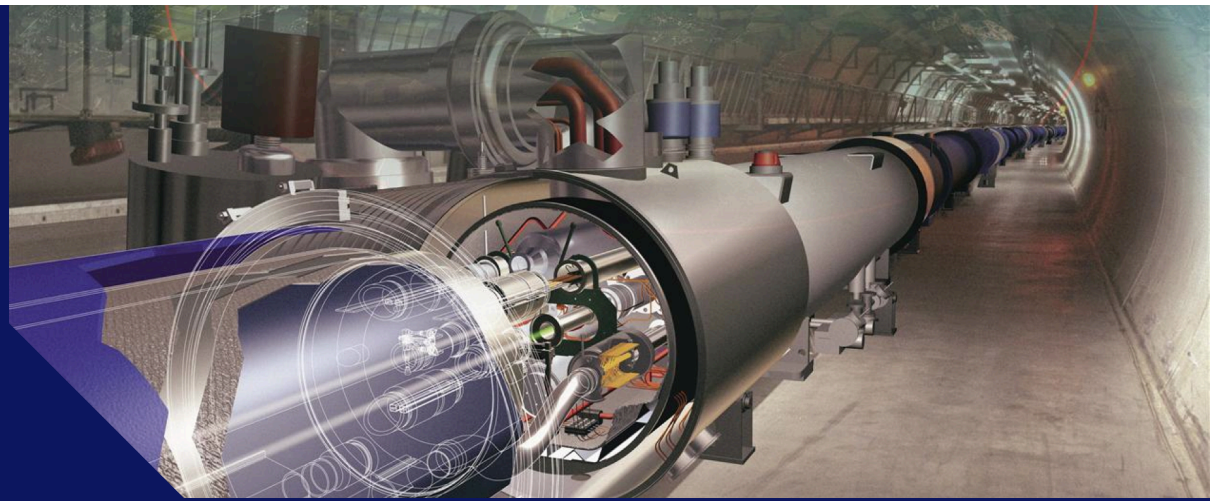


QCD@LHC 2019  
Buffalo, New York  
15 – 19 July 2019



## Recent developments with APPLGRID

Claire Gwenlan, Oxford

Mark Sutton, Sussex

Stefano Carrazza, CERN



# preface

- **theoretical uncertainties** (EG. proton pdfs, scales  $\mu_R$ ,  $\mu_F$ ) often **dominant systematic on many important physics processes at the LHC**
- searches for new massive particles; Higgs; fundamental SM parameters ( $\alpha_s$ , MW etc.)
- precision of data being matched by ever more sophisticated theoretical calcs. (at least NLO, and increasingly NNLO becomes available and also necessary)
- use cases, including usage of calculations in proton pdf fits, require running calculations many, many times with different pdfs, scales,  $\alpha_s$ , ...
- **practical usage limited due to long computation times (days, weeks, on large CPU farms)**

- **one solution:** store perturbative coefficients of (N)NLO QCD calculations in look up tables / grids

- **calculation needs to be run just once in full, to store coefficients;**

$$\frac{d\sigma}{dX} \sim \sum_{(i,j,p)} \int d\Gamma \alpha_s^p(Q_R^2) q_i(x_1, Q_F^2) q_j(x_2, Q_F^2) \frac{d\hat{\sigma}_{(p)}^{ij}}{dX}(x_1, x_2, Q_F^2, Q_R^2; S)$$

- **allows *a posteriori* convolution with any choice of pdf (and scales,  $\alpha_s$ , etc.)**

fast *a posteriori* convolution typically takes of **order milliseconds** rather than weeks  
typically reproduces original calculation to  **$10^{-4} - 10^{-5}$**  accuracy

# what is APPLGRID?

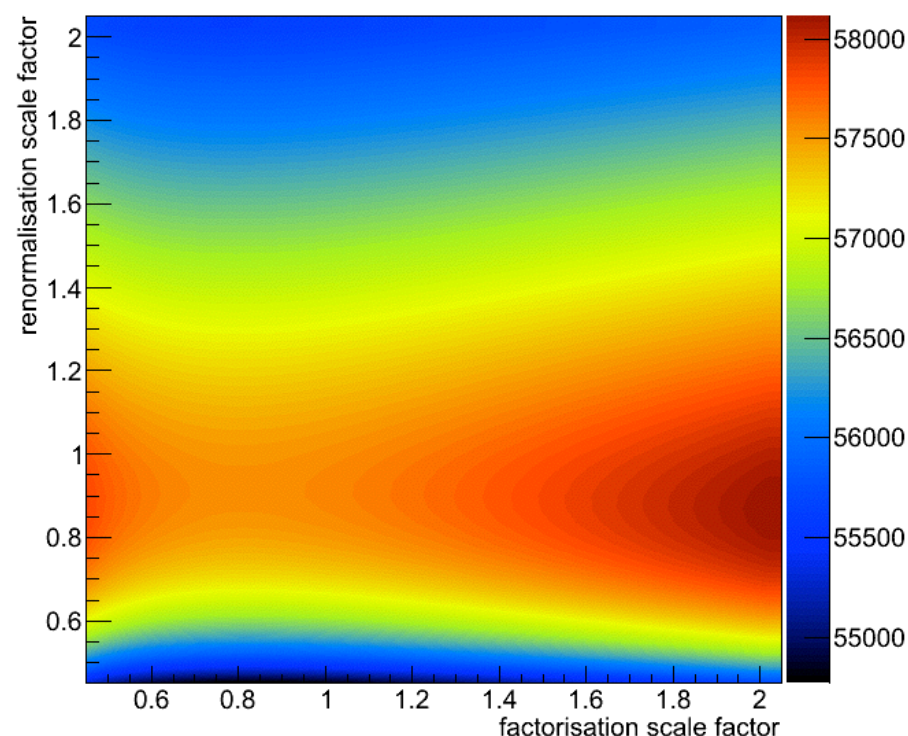
**APPLGRID**: open source package to build library of C++ utility classes for performing fast convolutions

interfaced to range of codes:

NLOjet++, MCFM, Sherpa, aMC@NLO, NNLOJET

allows fast reproduction of cross section calculations with pre-existing grids

- arbitrary pdfs; including allowing different pdfs for each target hadron
- arbitrary  $\mu_R$  and  $\mu_F$  scale variations
- any number of multiplicative corrections can be stored and applied
- **can include photon density**
- **arbitrary beam energy rescaling**  
(previously only  $\sqrt{s}$  scaling available)
- **grid remapping; and conversion of fastNLO to APPLgrid format**



**current full featured development version:**

applgrid-1.5.39

<https://applgrid.hepforge.org/downloads/applgrid-1.5.39.tgz>

# recap of numerical technique

- For a calculation of a cross section from  $m = 1 \dots N$  weights,  $w_m$ , from a Monte Carlo integration with momentum fraction  $x_m$ , form the product

$$\sum_m w(x_m) q(x_m)$$

- Can interpolate the function  $q(x_m) \dots$

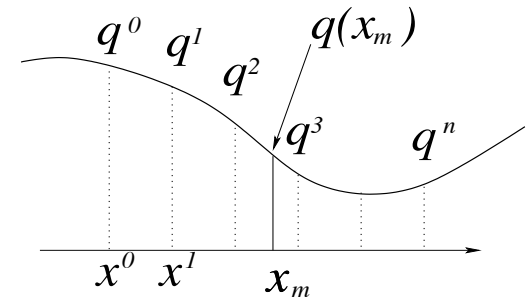
$$q(x_m) \approx \sum_i q^{(i)} I^{(i)}(x_m - x^{(i)})$$

- such that

$$\begin{aligned} \sum_m w(x_m) q(x_m) &\approx \sum_i q^{(i)} \sum_m w(x_m) I^{(i)}(x_m - x^{(i)}) \\ &\approx \sum_i q^{(i)} W^{(i)} \end{aligned}$$

- For a calculation of a cross section with  $m = 1 \dots N$  weights, from a Monte Carlo integration with momentum transfer  $Q^2$

$$\begin{aligned} d\sigma &= \sum_p \sum_{m=1}^N w_m^{(p)} \left( \frac{\alpha_s(Q_m^2)}{2\pi} \right)^p q(x_m, Q_m^2) \\ &= \sum_p \sum_{ij} q(x_{(i)}, Q_{(j)}^2) \left( \frac{\alpha_s(Q_{(j)}^2)}{2\pi} \right)^p \sum_m w_m^{(p)} I_i^x(x_m) I_j^{Q^2}(Q_m^2) \\ &= \sum_p \sum_{ij} q(x_{(i)}, Q_{(j)}^2) \left( \frac{\alpha_s(Q_{(j)}^2)}{2\pi} \right)^p W_{ij}^{(p)} \end{aligned}$$





# proton-proton collisions

- For pp collisions need an extra dimension for the PDF of the second colliding hadron

$$d\sigma = \sum_p \sum_{m=1}^N w_m^{(p)} \left( \frac{\alpha_s(Q_m^2)}{2\pi} \right)^p q_1(x_{1m}, Q_m^2) q_2(x_{2m}, Q_m^2)$$

- But there is an implicit summation over parton flavours. Make use of symmetries in the matrix elements to use a vector of  $k = 1 \dots M$  independent weights such that

$$\sum_{ij=q,\bar{q},g} w_{ij} q_{1i}(x_1) q_{2j}(x_2) = \sum_{k=1}^M w^{(k)} F^{(k)}(x_1, x_2)$$

- such that

$$d\sigma = \sum_p \sum_{k=1}^M \sum_{m=1}^N w_m^{(p)(k)} \left( \frac{\alpha_s(Q_m^2)}{2\pi} \right)^p F_m^{(k)}(x_{1m}, x_{2m}, Q_m^2)$$

- Which can be placed on a grid in the same way as for DIS

taken from talks by M. Sutton

everything is down to the **quality of interpolation of the pdf at the grid nodes**; pure quadratic technique which is not, in principle, subject to statistical fluctuation, **ie. each individual weight gets added to the grid, and should be well approximated individually**

applgrid is hosted by [Hepforge](#), IPPP Durham

*the APPLgrid project*

[Home](#) [Downloads](#) [Documentation](#) [Subversion](#) [Nightlies](#) [News](#) [Links](#)

### Convolution Code Download

Current version [applgrid-1.5.15](#)  
Basic example code [here](#)  
Standalone convolution utility [here](#)  
The [hoppet](#) code version 1.2.0 for QCD evolution from Gavin Salam and Juan Rojo.

### Calculation Code

MCFM: (use standard mcfm code)  
[mcfm-patch](#) (mcfm applgrid patch 0.0.8)  
[mcfm-bridge](#) (version 0.0.35 - for mcfm-6.8)  
NLOjet++:  
[nlojet++ 4.0.1](#) (applgrid version 0.0.2)  
[nlojet++ lhpdf wrapper 1.0.0](#) (applgrid version 0.0.2)  
[nlojet++ user module](#) (applgrid version 0.0.2)

### Grid Download

Full details on the [Downloads](#) page:

### Quick Start Guide

[how to run the APPLgrid code](#)

### Citation

Please cite the APPLgrid as  
[Eur Phys J C 66 \(2010\) 503](#)

## Welcome

The APPLgrid project provides a fast and flexible way to reproduce the results of full NLO calculations with any input parton distribution set in only a few milliseconds rather than the weeks normally required to gain adequate statistics.

Written in C++ (although a fortran interface is included) it can be used for the calculation of any process where the hard subprocess weights from the convolution with the PDF are available from the calculation.

The user can use existing grids simply to obtain the fast cross sections, as with fastNLO, but the complete project is publicly available should the user wish to generate the grids themselves for new cross sections. In this case, the user interfaces the grid code with NLO calculation, and after running the NLO calculation to achieve the required statistical precision once, the results of the calculation with any different parton distribution set can be calculated, typically in around 1 to 100ms, depending on the size of the grid.

At present, examples exist for MCFM and nlojet++. Since the user code that needs to run to extract the weights and create the grid may require changes to the NLO calculation code or may be dependent on specific versions of the code, the specific versions of both MCFM and nlojet++ are included here

An interface to fastNLO (version 1) grids is included.

The code is under continuous development so please come back soon for more information.

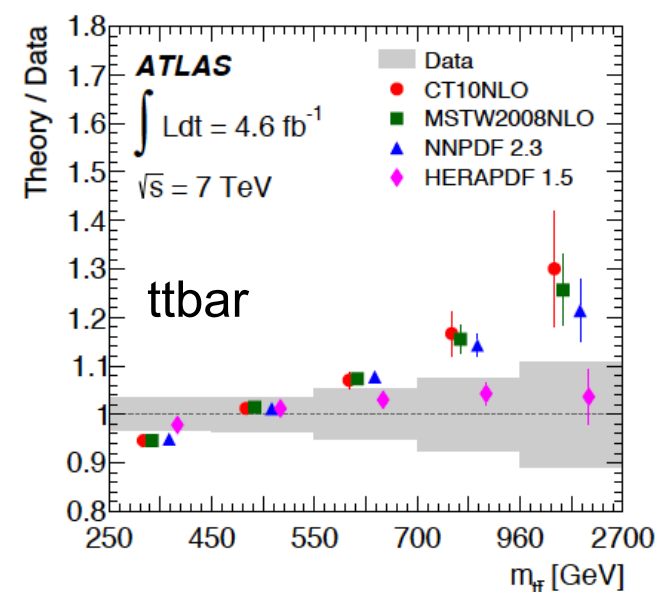
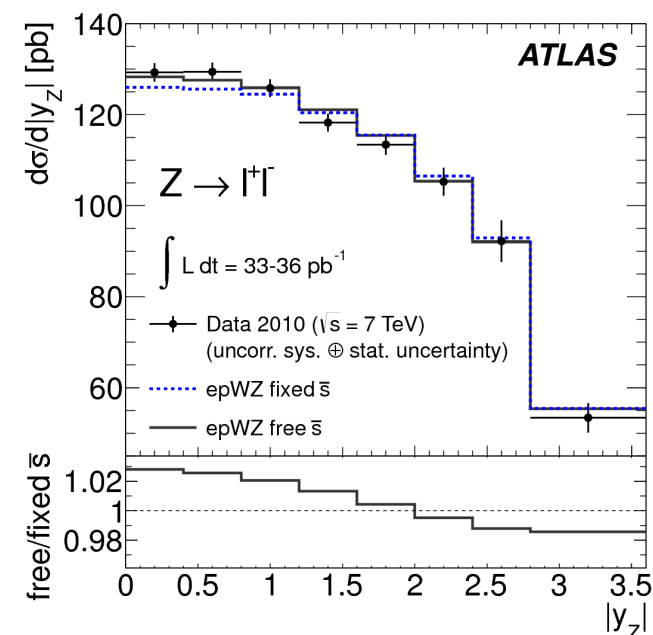
Mark Sutton, Pavel Starovoitov, Tancredi Carli, Claire Gwenlan and Gavin Salam - send mail to the authors: [applgrid@projects.hepforge.org](mailto:applgrid@projects.hepforge.org): Last updated Sun 9 Dec 2018 16:53:21

# recap: available processes at NLO

- essentially all of NLO QCD available with **APPLGRID** (and fastNLO)
- currently available interfaces for:
  - NLOJet++**: jet production
  - MCFM**: electroweak boson, heavy flavour, V+Jet; plus generic interface for all remaining processes
  - aMC@NLO**: all fixed order NLO processes, using aMCfast, arXiv:[1406.7693](https://arxiv.org/abs/1406.7693)
  - Sherpa**: fixed NLO, using MCgrid, arXiv:[1312.4460](https://arxiv.org/abs/1312.4460); plus native APPLGRID-Sherpa interface

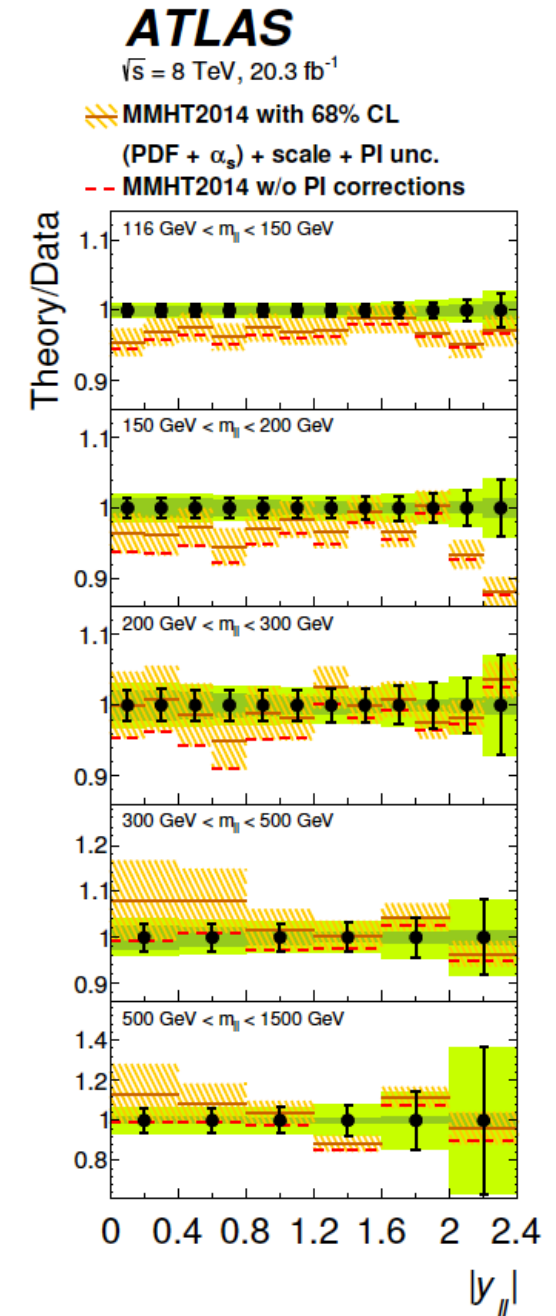
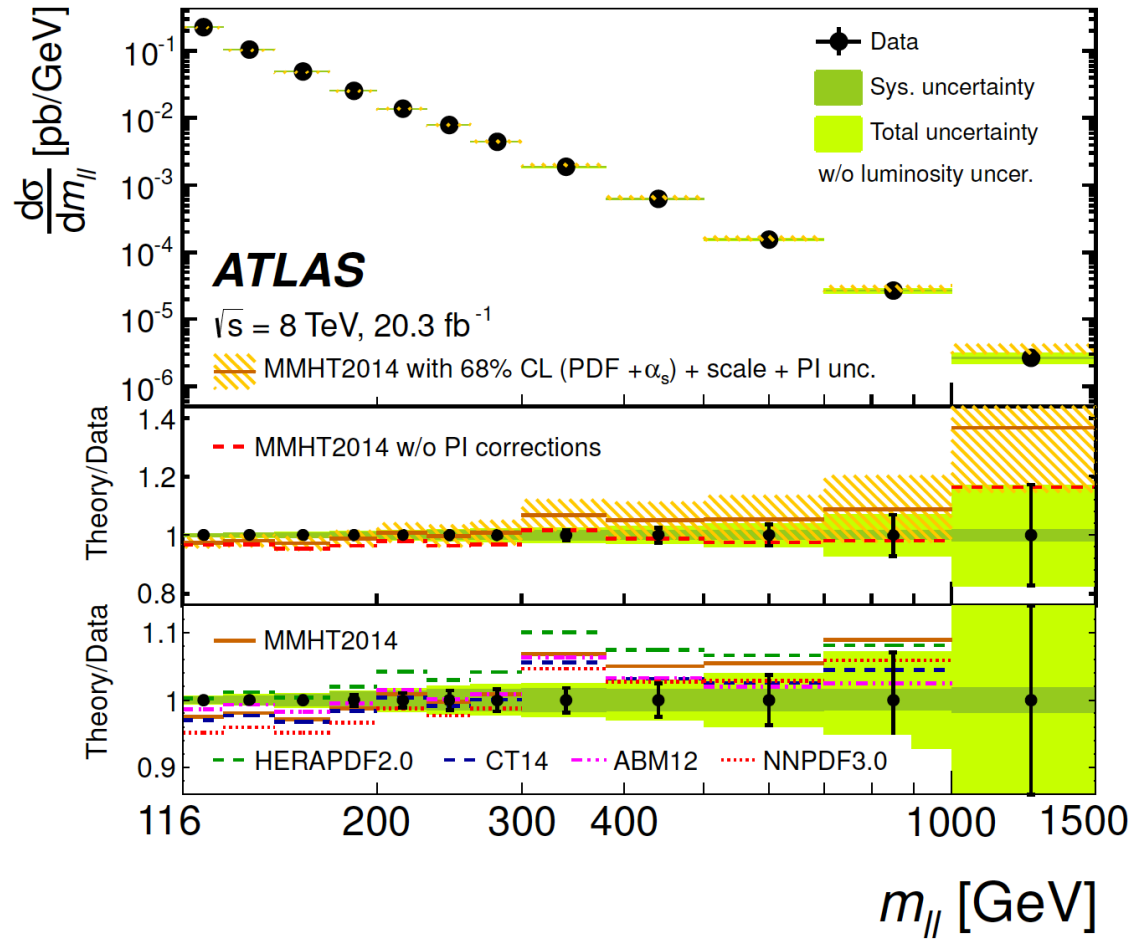
NB, much emphasis now on active development of **APPLfast framework for use with NNLO calculations**; common project with NNLOJET and [fastNLO](https://arxiv.org/abs/1406.7693) authors (see talk by K.Rabbertz, this conference)

**APPLGRID** development still nevertheless active ...



# photon induced processes

ATLAS coll., arXiv: [1606.01736](https://arxiv.org/abs/1606.01736)



# photon induced processes

**APPLGRID modifications to allow photon to be treated like any other parton in the proton;** currently use a modification of the lhaglu-type LHAPDF interface

```
extern "C" void evolvepdf(const double& x, const double& Q, double* xfx) {
    xfx += 6;
    for (int i = -6; i < 7; i++) xfx[i] = pdf->xfxQ(i, x, Q);
    xfx[7] = pdf->xfxQ(22, x, Q);
}
```

native LHAPDF6 calls, with call to access photon density in lhaglu type wrapper

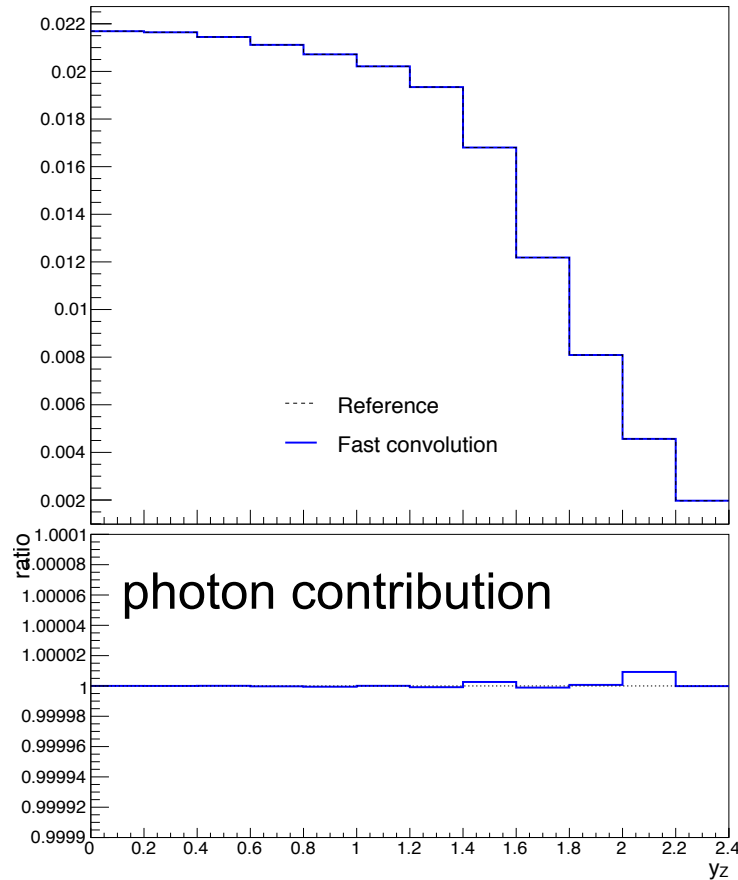
APPLGRID intrinsic lumi\_pdf allows additional parton, EG. 7 or 22 for the photon

0						
0	2	2	-2	4	-4	
1	2	0	-4	0	-2	
2	2	2	0	4	0	
3	3	1	-1	3	-3	5 -5
4	3	0	-5	0	-3	0 -1
5	3	1	0	3	0	5 0
6	2	-4	4	-2	2	
7	2	0	2	0	4	
8	2	-4	0	-2	0	
9	3	-5	5	-3	3	-1 1
10	3	0	1	0	3	0 5
11	3	-5	0	-3	0	-1 0
12	1	7	7			



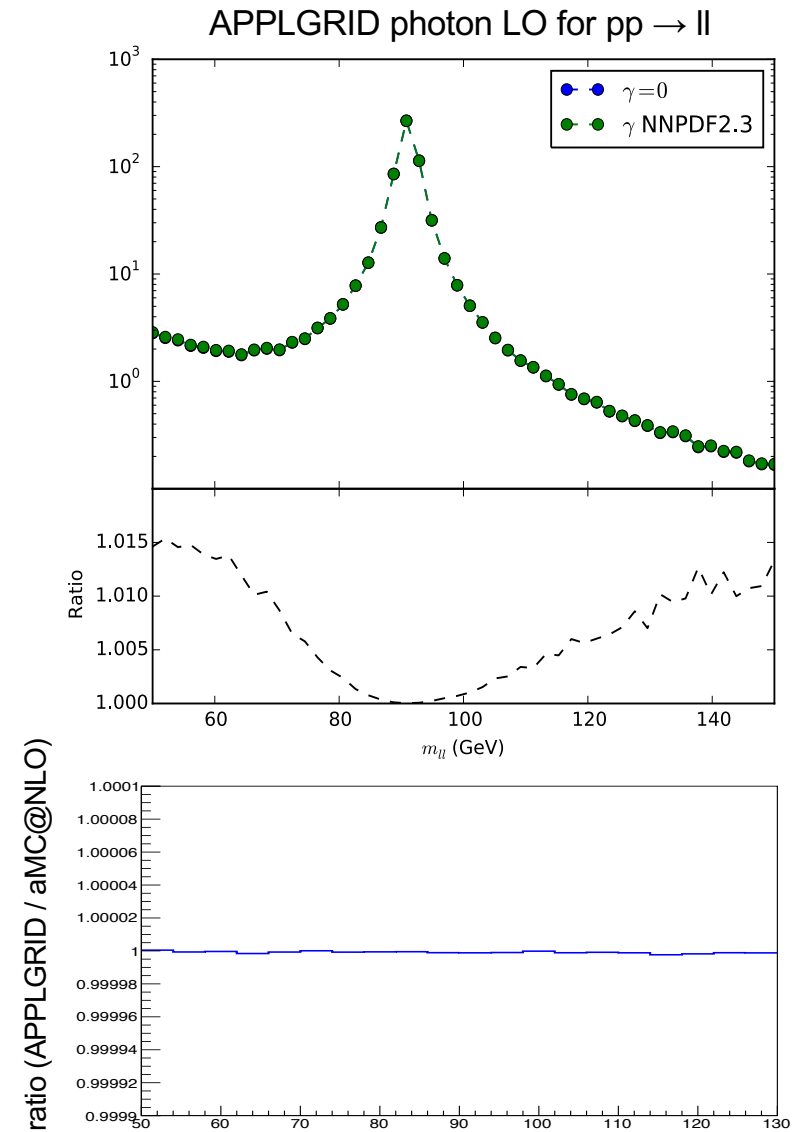
pdf : config nprocesses: 13		
0	0	$(u, \bar{u}) + (c, \bar{c})$
1	1	$(g, \bar{c}) + (g, \bar{u})$
2	2	$(u, g) + (c, g)$
3	3	$(d, \bar{d}) + (s, \bar{s}) + (b, \bar{b})$
4	4	$(g, \bar{b}) + (g, \bar{s}) + (g, \bar{d})$
5	5	$(d, g) + (s, g) + (b, g)$
6	6	$(\bar{c}, c) + (\bar{u}, u)$
7	7	$(g, u) + (g, c)$
8	8	$(\bar{c}, g) + (\bar{u}, g)$
9	9	$(\bar{b}, b) + (\bar{s}, s) + (\bar{d}, d)$
10	10	$(g, d) + (g, s) + (g, b)$
11	11	$(\bar{b}, g) + (\bar{s}, g) + (\bar{d}, g)$
12	12	$(\gamma, \gamma)$

# photon induced processes



## photon contribution to the proton

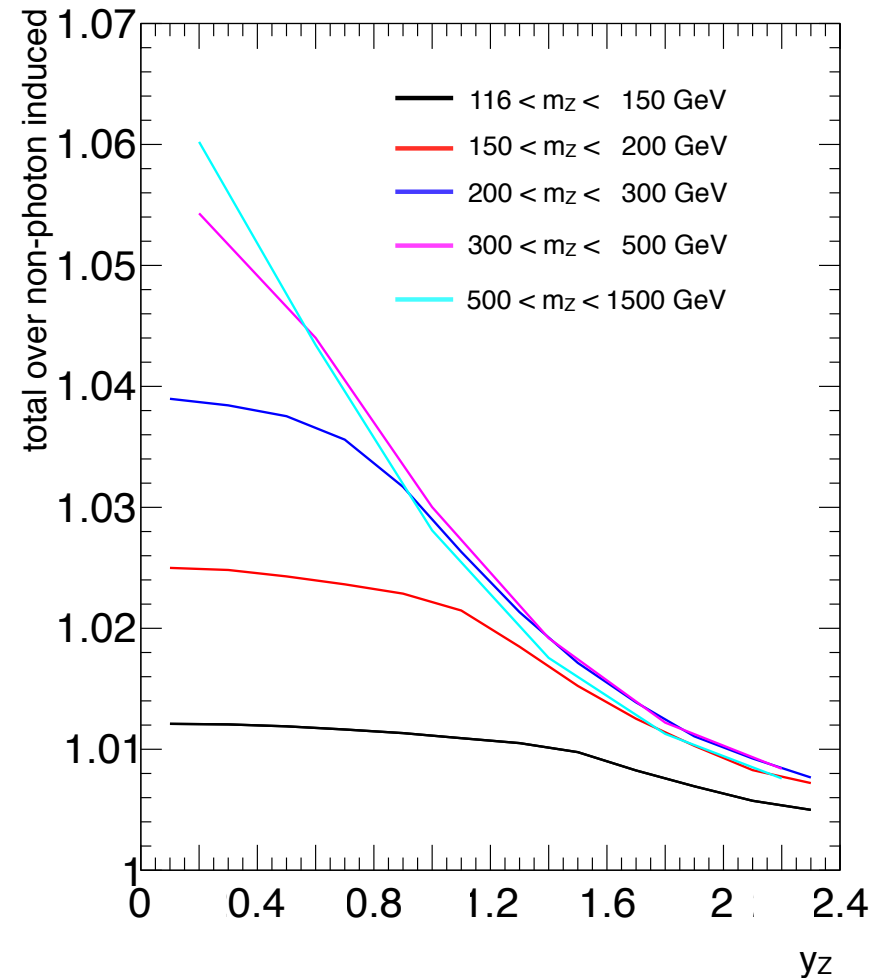
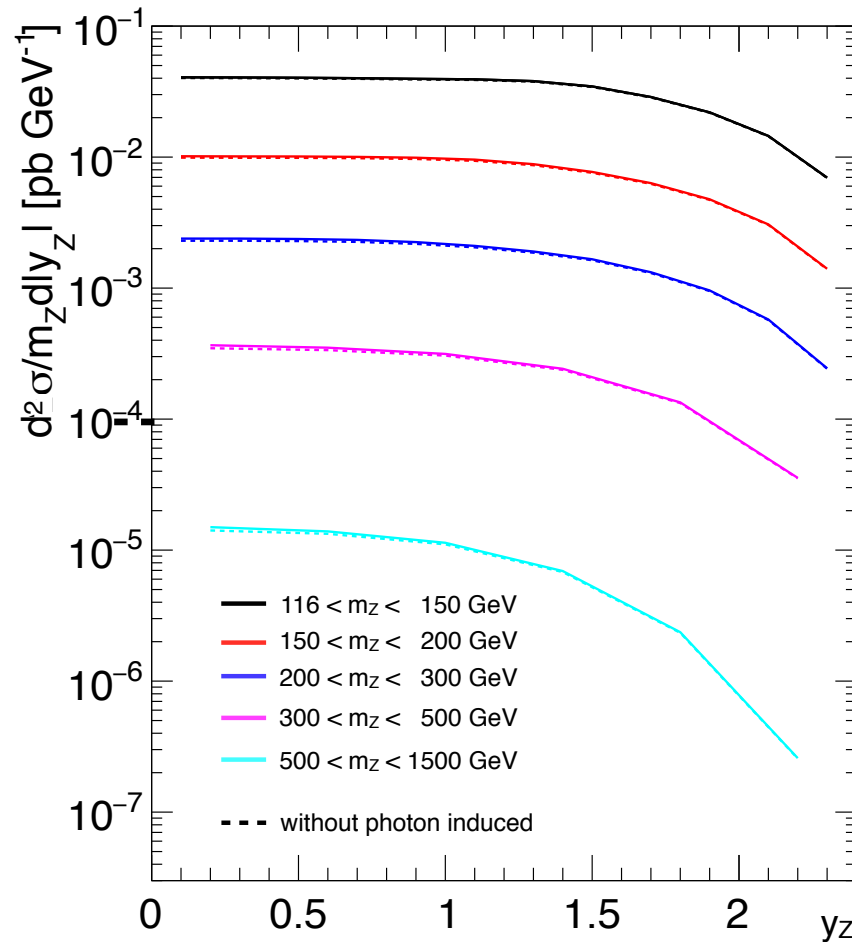
original implementation in APPLGRID private release, some time ago (S. Carrazza), EG. as used in arXiv: [1712.07053](https://arxiv.org/abs/1712.07053)



now code fully implemented into main development branch of APPLGRID

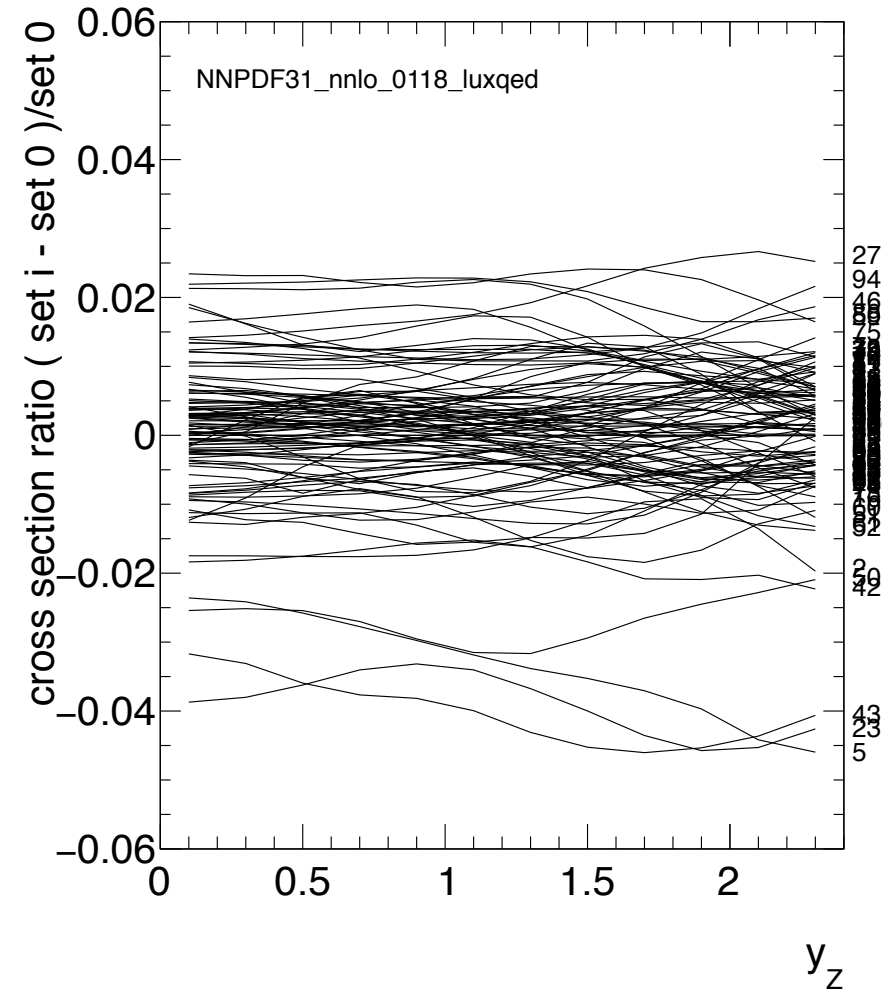
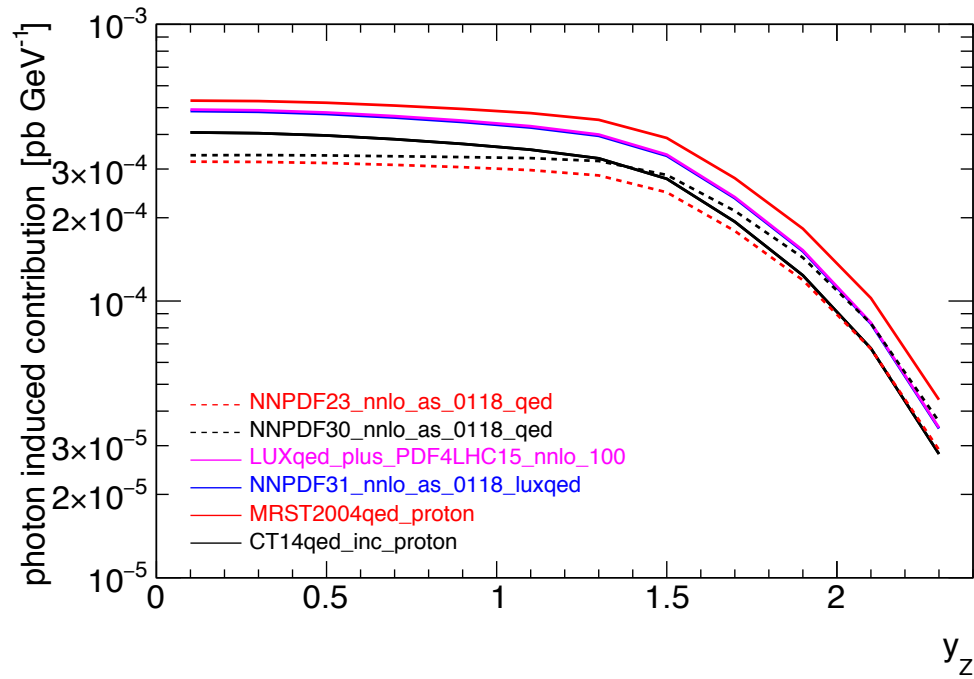


# photon induced processes



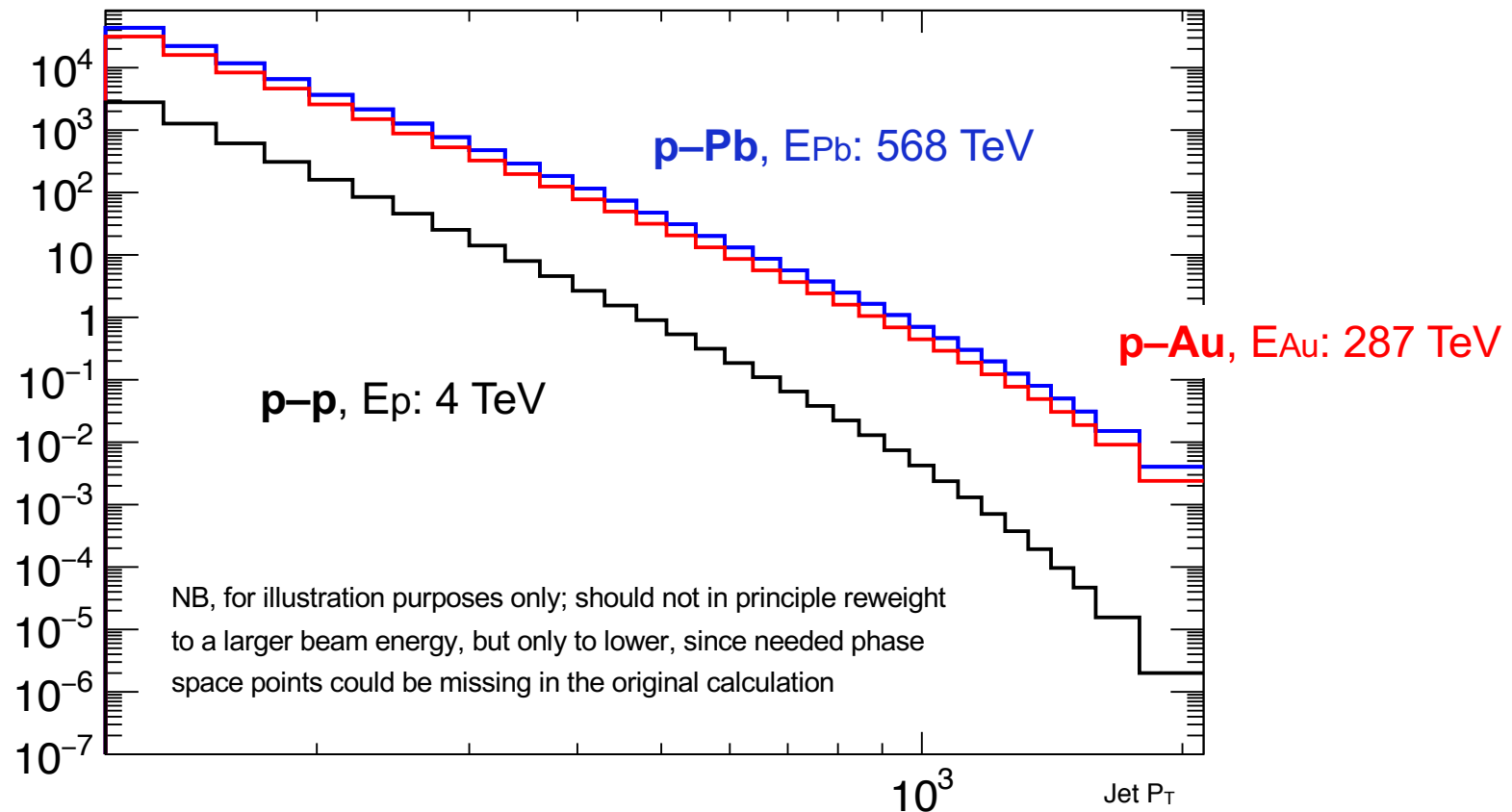
- photon induced contribution generally small, but can be several percent at large mass (c.f. HM DY measurement from ATLAS, arXiv:[1606.01736](https://arxiv.org/abs/1606.01736))

# photon induced processes



- fast reproduction of the cross sections with different PDFs

## use for heavy ion collisions (etc.)

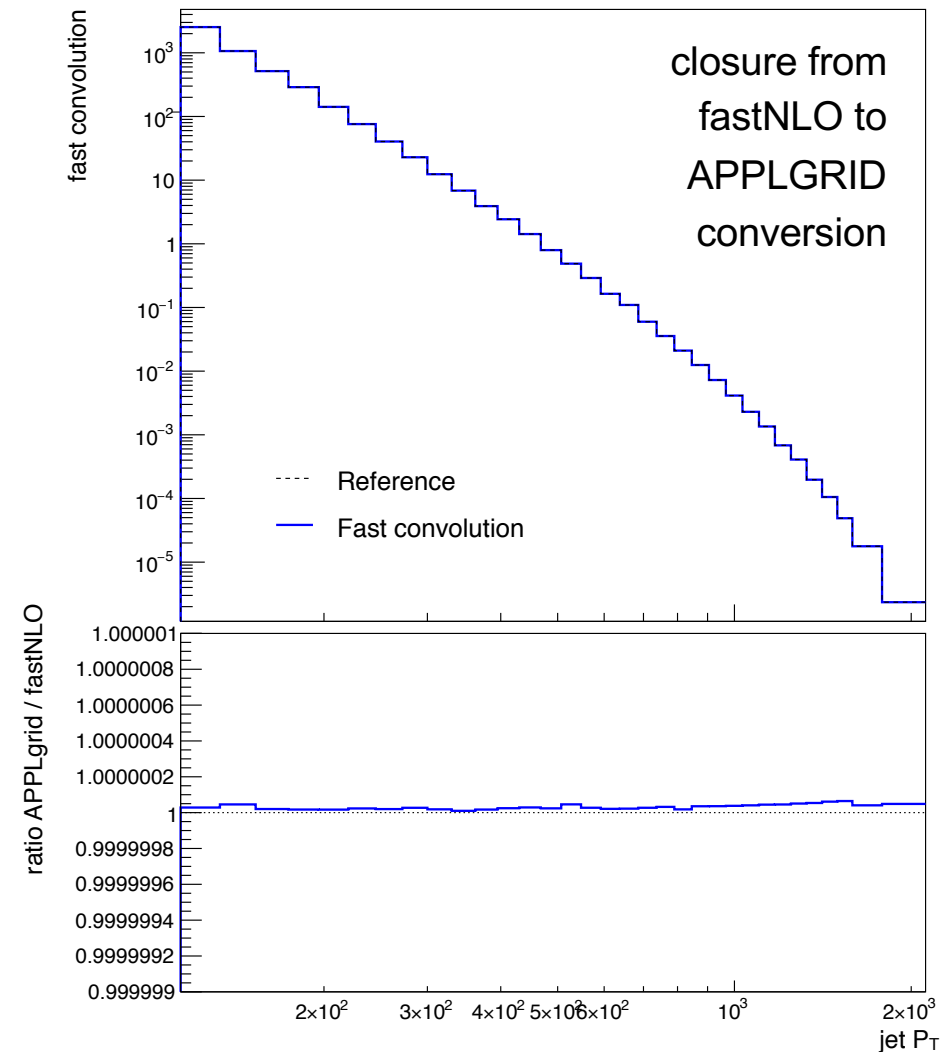


- can perform a posteriori convolution using **different pdfs** for each target hadron
  - can also **vary beam energy** independently for each beam
- (cf. previously, a  **$\sqrt{s}$  scaling** only was implemented)

```
double invbeam1 = 1;  
double invbeam2 = 4/287;  
  
TH1D* h = g.convolute( evolvepdf, evolvepdfi, alphaspdf,  
                        nloop, rscale, fscale,  
                        invbeam1, invbeam2 );
```

# grid remapping

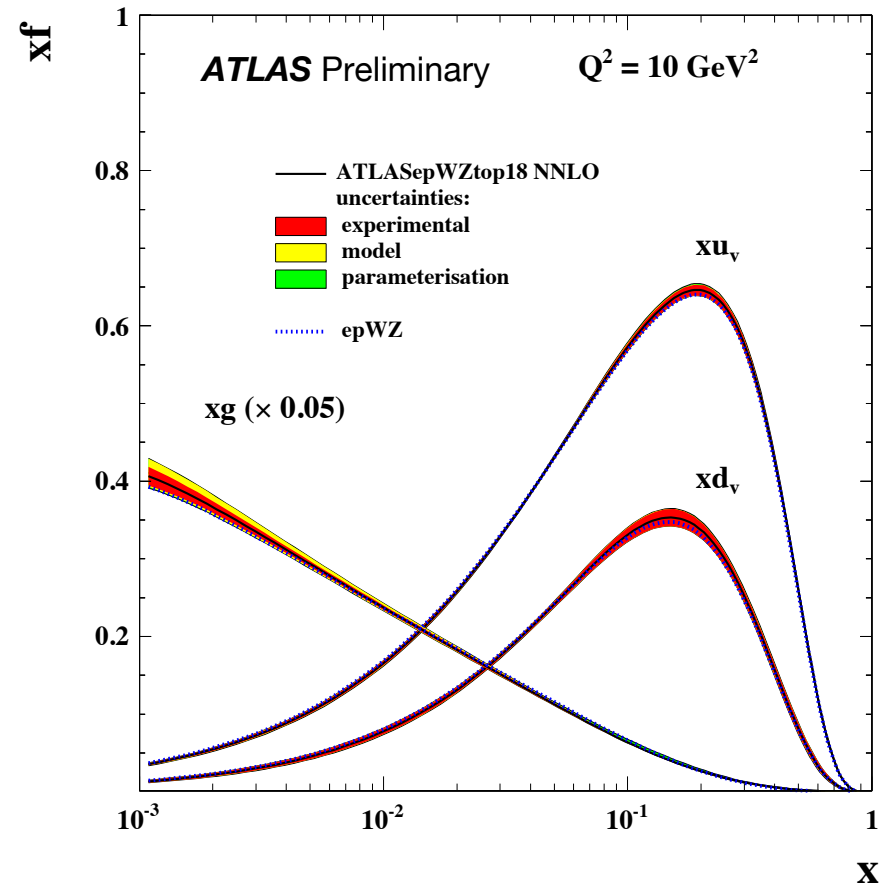
- in principle, can treat weights on a grid at specific grid nodes exactly as the weights from an independent calculation
- **can fill the weights onto a new grid with different parameters:** different number of grid nodes; different transform for  $x, Q^2$ ; different scales etc.
- **can be used to reduce size of grids;** a posteriori find the smallest grid size for any given required precision
- **code implemented to remap fastNLO tables to APPLGRID format** →
- significant saving in grid file size and memory footprint due to APPLGRID sparse structure etc.



# grids for top production at NNLO

- **top cross sections at NNLO QCD**  
available for some time from Czakon et al. in **fastNLO** format, [arXiv:1704.08551](https://arxiv.org/abs/1704.08551)
- <http://www.precision.hep.phy.cam.ac.uk/results/ttbar-fastnlo>
- used in EG. **ATLASepWZtop18fit**,  
[ATL-PHYS-PUB-2018-017](https://arxiv.org/abs/1805.08107)

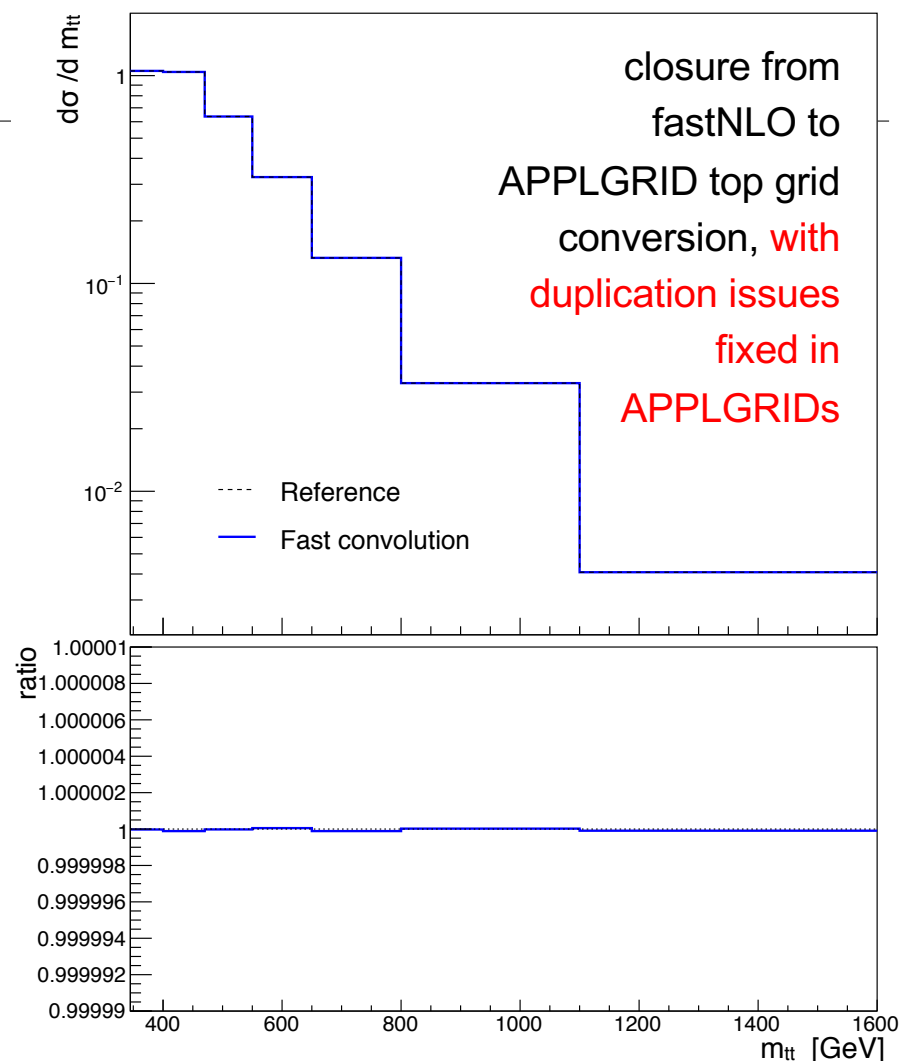
(code is closed source; not possible to see how grid filling has been implemented)



these have now been converted to APPLGRID format

# grids for top production at NNLO

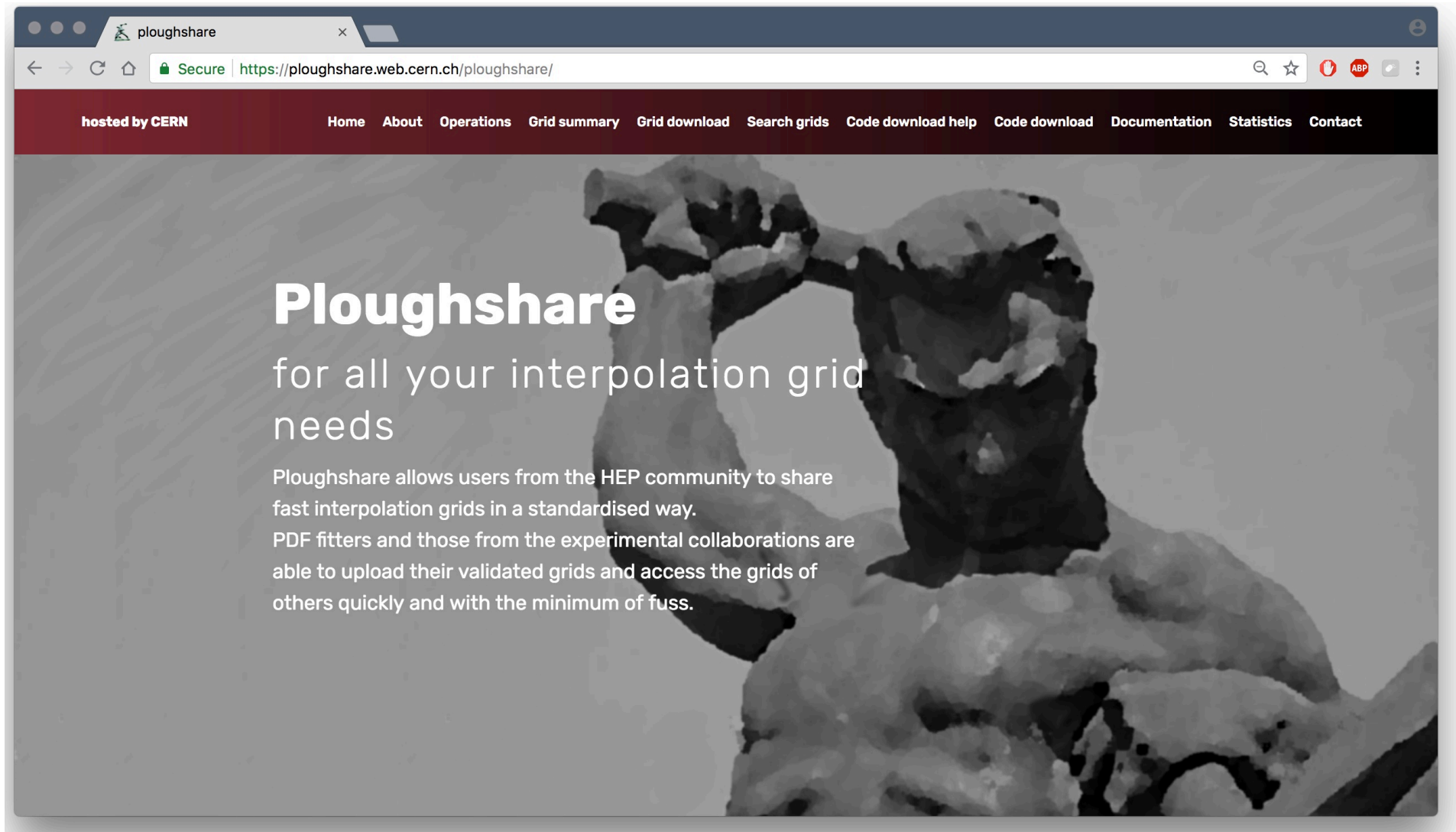
- APPLGRIDs able to reproduce fastNLO tables to within single floating precision
- NB, fastNLO tables stored in text format are accurate only to single precision ( $\sim 7$ s.f.) with small modification to APPLGRID code, would be possible to reproduce fastNLO tables exactly to double precision (as we have previously with v1), but we see no urgent need
- APPLGRIDs smaller, with reduced memory footprint and faster convolution
- not (primarily) due to APPLGRID format per se, but due to **non-optimal implementation of grid filling in provided fastNLO tables**; duplication of parton-parton luminosities, EG at NLO, basic  $11 \times 11$  used, but then duplicated by at least a factor of 2; **fixed in APPLGRID versions**



APPLGRIDs already available upon request, and on ploughshare soon...



[ploughshare.web.cern.ch](https://ploughshare.web.cern.ch)



- repository for download and, as a registered user, upload of grids (APPLgrid and fastNLO)

# outlook

- **APPLGRID project** is increasingly mature [applgrid.hepforge.org](http://applgrid.hepforge.org)
- emphasis now mostly looking towards development for use with **NNLO QCD cross section calculations and interface with NNLOJET and fastNLO (APPLfast project)**
- **APPLGRID development still nevertheless active, with desirable features still being added**
- in discussion with developers from other packages, hope to move towards more standardised interfaces with other code
- always open to new suggestions or requests, especially if people are willing to contribute to the project

extras

# top parton luminosities at LO

pdf : mass-ttbar-pT-t-slice-350.000000-550.000000-18-LO.config nprocesses: 18

---

0	0	$(\bar{b}, \bar{b}) + (\bar{c}, \bar{c}) + (\bar{s}, \bar{s}) + (\bar{u}, \bar{u}) + (\bar{d}, \bar{d}) + (d, d) + (u, u) + (s, s) + (c, c) + (b, b)$
1	1	$(\bar{c}, \bar{b}) + (\bar{s}, \bar{b}) + (\bar{u}, \bar{b}) + (\bar{d}, \bar{b}) + (\bar{b}, \bar{c}) + (\bar{s}, \bar{c}) + (\bar{u}, \bar{c}) + (\bar{d}, \bar{c}) + (\bar{b}, \bar{s}) + (\bar{c}, \bar{s}) + (\bar{u}, \bar{s}) + (\bar{d}, \bar{s}) + (\bar{b}, \bar{u}) + (\bar{c}, \bar{u}) + (\bar{s}, \bar{u}) + (\bar{d}, \bar{u}) + (\bar{b}, \bar{d}) + (\bar{c}, \bar{d}) + (\bar{s}, \bar{d}) + (\bar{u}, \bar{d}) + (u, d) + (s, d) + (c, d) + (b, d) + (d, u) + (s, u) + (c, u) + (b, u) + (d, s) + (u, s) + (c, s) + (b, s) + (d, c) + (u, c) + (s, c) + (b, c) + (d, b) + (u, b) + (s, b) + (c, b)$
2	2	$(d, \bar{b}) + (u, \bar{b}) + (s, \bar{b}) + (c, \bar{b}) + (d, \bar{c}) + (u, \bar{c}) + (s, \bar{c}) + (b, \bar{c}) + (d, \bar{s}) + (u, \bar{s}) + (c, \bar{s}) + (b, \bar{s}) + (d, \bar{u}) + (s, \bar{u}) + (c, \bar{u}) + (b, \bar{u}) + (u, \bar{d}) + (s, \bar{d}) + (c, \bar{d}) + (b, \bar{d})$
3	3	$(b, \bar{b}) + (c, \bar{c}) + (s, \bar{s}) + (u, \bar{u}) + (d, \bar{d})$
4	4	$(g, \bar{b}) + (g, \bar{c}) + (g, \bar{s}) + (g, \bar{u}) + (g, \bar{d}) + (g, d) + (g, u) + (g, s) + (g, c) + (g, b)$
5	5	$(\bar{b}, d) + (\bar{c}, d) + (\bar{s}, d) + (\bar{u}, d) + (\bar{b}, u) + (\bar{c}, u) + (\bar{s}, u) + (\bar{d}, u) + (\bar{b}, s) + (\bar{c}, s) + (\bar{u}, s) + (\bar{d}, s) + (\bar{b}, c) + (\bar{s}, c) + (\bar{u}, c) + (\bar{d}, c) + (\bar{c}, b) + (\bar{s}, b) + (\bar{u}, b) + (\bar{d}, b)$
6	6	$(\bar{d}, d) + (\bar{u}, u) + (\bar{s}, s) + (\bar{c}, c) + (\bar{b}, b)$
7	7	$(\bar{b}, g) + (\bar{c}, g) + (\bar{s}, g) + (\bar{u}, g) + (\bar{d}, g) + (d, g) + (u, g) + (s, g) + (c, g) + (b, g)$
8	8	$(g, g)$
9	9	$(\bar{b}, \bar{b}) + (\bar{c}, \bar{c}) + (\bar{s}, \bar{s}) + (\bar{u}, \bar{u}) + (\bar{d}, \bar{d}) + (d, d) + (u, u) + (s, s) + (c, c) + (b, b)$
10	10	$(\bar{c}, \bar{b}) + (\bar{s}, \bar{b}) + (\bar{u}, \bar{b}) + (\bar{d}, \bar{b}) + (\bar{b}, \bar{c}) + (\bar{s}, \bar{c}) + (\bar{u}, \bar{c}) + (\bar{d}, \bar{c}) + (\bar{b}, \bar{s}) + (\bar{c}, \bar{s}) + (\bar{u}, \bar{s}) + (\bar{d}, \bar{s}) + (\bar{b}, \bar{u}) + (\bar{c}, \bar{u}) + (\bar{s}, \bar{u}) + (\bar{d}, \bar{u}) + (\bar{b}, \bar{d}) + (\bar{c}, \bar{d}) + (\bar{s}, \bar{d}) + (\bar{u}, \bar{d}) + (u, d) + (s, d) + (c, d) + (b, d) + (d, u) + (s, u) + (c, u) + (b, u) + (d, s) + (u, s) + (c, s) + (b, s) + (d, c) + (u, c) + (s, c) + (b, c) + (d, b) + (u, b) + (s, b) + (c, b)$
11	11	$(d, \bar{b}) + (u, \bar{b}) + (s, \bar{b}) + (c, \bar{b}) + (d, \bar{c}) + (u, \bar{c}) + (s, \bar{c}) + (b, \bar{c}) + (d, \bar{s}) + (u, \bar{s}) + (c, \bar{s}) + (b, \bar{s}) + (d, \bar{u}) + (s, \bar{u}) + (c, \bar{u}) + (b, \bar{u}) + (u, \bar{d}) + (s, \bar{d}) + (c, \bar{d}) + (b, \bar{d})$
12	12	$(b, \bar{b}) + (c, \bar{c}) + (s, \bar{s}) + (u, \bar{u}) + (d, \bar{d})$
13	13	$(g, \bar{b}) + (g, \bar{c}) + (g, \bar{s}) + (g, \bar{u}) + (g, \bar{d}) + (g, d) + (g, u) + (g, s) + (g, c) + (g, b)$
14	14	$(\bar{b}, d) + (\bar{c}, d) + (\bar{s}, d) + (\bar{u}, d) + (\bar{b}, u) + (\bar{c}, u) + (\bar{s}, u) + (\bar{d}, u) + (\bar{b}, s) + (\bar{c}, s) + (\bar{u}, s) + (\bar{d}, s) + (\bar{b}, c) + (\bar{s}, c) + (\bar{u}, c) + (\bar{d}, c) + (\bar{c}, b) + (\bar{s}, b) + (\bar{u}, b) + (\bar{d}, b)$
15	15	$(\bar{d}, d) + (\bar{u}, u) + (\bar{s}, s) + (\bar{c}, c) + (\bar{b}, b)$
16	16	$(\bar{b}, g) + (\bar{c}, g) + (\bar{s}, g) + (\bar{u}, g) + (\bar{d}, g) + (d, g) + (u, g) + (s, g) + (c, g) + (b, g)$
17	17	$(g, g)$

---

as appearing in current fastNLO top grids (note duplications; **fixed in APPLGRID conversions**)

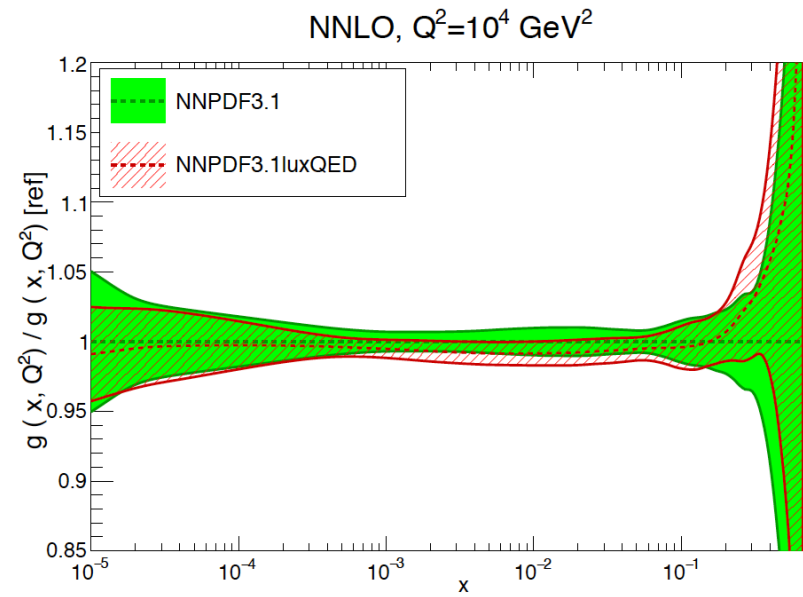
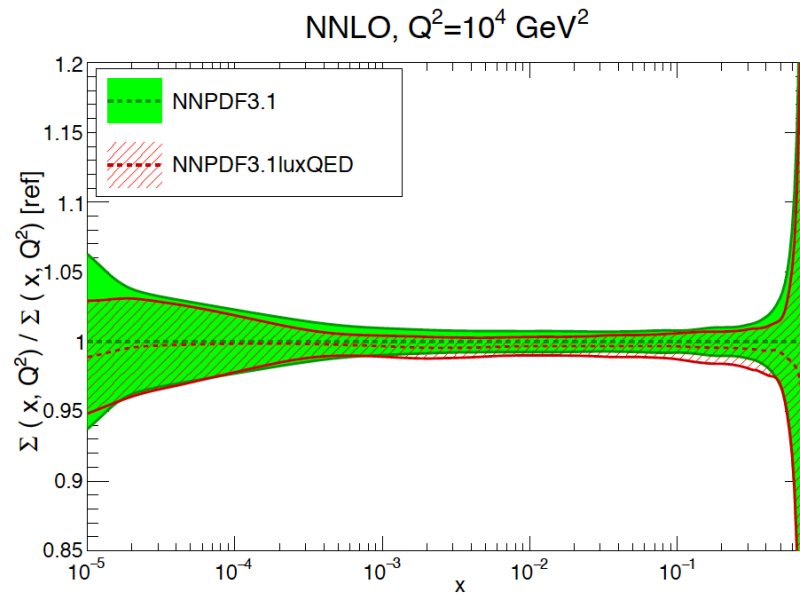
# top parton luminosities at NLO

pdf : mass-ttbar-pT-t-slice-350.000000-550.000000-18-NLO.config nprocesses: 260

0	0	$(\bar{b}, \bar{b}) + (\bar{c}, \bar{c}) + (\bar{s}, \bar{s}) + (\bar{u}, \bar{u}) + (\bar{d}, \bar{d}) + (d, d) + (u, u) + (s, s) + (c, c) + (b, b)$	
1	1	$(\bar{c}, \bar{b}) + (\bar{s}, \bar{b}) + (\bar{u}, \bar{b}) + (\bar{d}, \bar{b}) + (\bar{b}, \bar{c}) + (\bar{s}, \bar{c}) + (\bar{u}, \bar{c}) + (\bar{d}, \bar{c}) + (\bar{b}, \bar{s}) + (\bar{c}, \bar{s}) + (\bar{u}, \bar{s}) + (\bar{d}, \bar{s}) + (\bar{b}, \bar{u}) + (\bar{c}, \bar{u}) + (\bar{s}, \bar{u}) + (\bar{d}, \bar{u}) + (\bar{b}, \bar{d}) + (\bar{c}, \bar{d}) + (\bar{s}, \bar{d}) + (\bar{u}, \bar{d}) + (u, d) + (s, d) + (c, d) + (b, d) + (d, u) + (s, u) + (c, u) + (b, u) + (d, s) + (u, s) + (c, s) + (b, s) + (d, c) + (u, c) + (s, c) + (b, c) + (d, b) + (u, b) + (s, b) + (c, b)$	
2	2	$(d, \bar{b}) + (u, \bar{b}) + (s, \bar{b}) + (c, \bar{b}) + (d, \bar{c}) + (u, \bar{c}) + (s, \bar{c}) + (b, \bar{c}) + (d, \bar{s}) + (u, \bar{s}) + (c, \bar{s}) + (b, \bar{s}) + (d, \bar{u}) + (s, \bar{u}) + (c, \bar{u}) + (b, \bar{u}) + (u, \bar{d}) + (s, \bar{d}) + (c, \bar{d}) + (b, \bar{d})$	
3	3	$(b, \bar{b}) + (c, \bar{c}) + (s, \bar{s}) + (u, \bar{u}) + (d, \bar{d})$	
4	4	$(g, \bar{b}) + (g, \bar{c}) + (g, \bar{s}) + (g, \bar{u}) + (g, \bar{d}) + (g, d) + (g, u) + (g, s) + (g, c) + (g, b)$	
5	5	$(\bar{b}, d) + (\bar{c}, d) + (\bar{s}, d) + (\bar{u}, d) + (\bar{b}, u) + (\bar{c}, u) + (\bar{s}, u) + (\bar{d}, u) + (\bar{b}, s) + (\bar{c}, s) + (\bar{u}, s) + (\bar{d}, s) + (\bar{b}, c) + (\bar{s}, c) + (\bar{u}, c) + (\bar{d}, c) + (\bar{c}, b) + (\bar{s}, b) + (\bar{u}, b) + (\bar{d}, b)$	
6	6	$(d, d) + (\bar{u}, u) + (\bar{s}, s) + (\bar{c}, c) + (\bar{b}, b)$	
7	7	$(\bar{b}, g) + (\bar{c}, g) + (\bar{s}, g) + (\bar{u}, g) + (\bar{d}, g) + (d, g) + (u, g) + (s, g) + (c, g) + (b, g)$	
8	8	$(g, g)$	
9	9	127	127 $(c, g)$
10	10	128	128 $(b, g)$
11	11	129	129 $(g, g)$
12	12	130	130 $(\bar{b}, \bar{b}) + (\bar{c}, \bar{c}) + (\bar{s}, \bar{s}) + (\bar{u}, \bar{u}) + (\bar{d}, \bar{d}) + (d, d) + (u, u) + (s, s) + (c, c) + (b, b)$
13	13	131	131 $(\bar{c}, \bar{b}) + (\bar{s}, \bar{b}) + (\bar{u}, \bar{b}) + (\bar{d}, \bar{b}) + (\bar{b}, \bar{c}) + (\bar{s}, \bar{c}) + (\bar{u}, \bar{c}) + (\bar{d}, \bar{c}) + (\bar{b}, \bar{s}) + (\bar{c}, \bar{s}) + (\bar{u}, \bar{s}) + (\bar{d}, \bar{s}) + (\bar{b}, \bar{u}) + (\bar{c}, \bar{u}) + (\bar{s}, \bar{u}) + (\bar{d}, \bar{u}) + (\bar{b}, \bar{d}) + (\bar{c}, \bar{d}) + (\bar{s}, \bar{d}) + (\bar{u}, \bar{d}) + (u, d) + (s, d) + (c, d) + (b, d) + (d, u) + (s, u) + (c, u) + (b, u) + (d, s) + (u, s) + (c, s) + (b, s) + (d, c) + (u, c) + (s, c) + (b, c) + (d, b) + (u, b) + (s, b) + (c, b)$
14	14		
15	15	132	132 $(d, \bar{b}) + (u, \bar{b}) + (s, \bar{b}) + (c, \bar{b}) + (d, \bar{c}) + (u, \bar{c}) + (s, \bar{c}) + (b, \bar{c}) + (d, \bar{s}) + (u, \bar{s}) + (c, \bar{s}) + (b, \bar{s}) + (d, \bar{u}) + (s, \bar{u}) + (c, \bar{u}) + (b, \bar{u}) + (u, \bar{d}) + (s, \bar{d}) + (c, \bar{d}) + (b, \bar{d})$
16	16		
17	17	133	133 $(b, \bar{b}) + (c, \bar{c}) + (s, \bar{s}) + (u, \bar{u}) + (d, \bar{d})$
18	18	134	134 $(g, \bar{b}) + (g, \bar{c}) + (g, \bar{s}) + (g, \bar{u}) + (g, \bar{d}) + (g, d) + (g, u) + (g, s) + (g, c) + (g, b)$
19	19	135	135 $(\bar{b}, d) + (\bar{c}, d) + (\bar{s}, d) + (\bar{u}, d) + (\bar{b}, u) + (\bar{c}, u) + (\bar{s}, u) + (\bar{d}, u) + (\bar{b}, s) + (\bar{c}, s) + (\bar{u}, s) + (\bar{d}, s) + (\bar{b}, c) + (\bar{s}, c) + (\bar{u}, c) + (\bar{d}, c) + (\bar{c}, b) + (\bar{s}, b) + (\bar{u}, b) + (\bar{d}, b)$
20	20		
21	21	136	136 $(d, d) + (\bar{u}, u) + (\bar{s}, s) + (\bar{c}, c) + (\bar{b}, b)$
22	22	137	137 $(\bar{b}, g) + (\bar{c}, g) + (\bar{s}, g) + (\bar{u}, g) + (\bar{d}, g) + (d, g) + (u, g) + (s, g) + (c, g) + (b, g)$
23	23	138	138 $(g, g)$
24	24	139	139 $(\bar{b}, \bar{b})$
25	25	140	140 $(\bar{c}, \bar{b})$
26	26	$(u, \bar{c})$	
27	27	$(s, \bar{c})$	
28	28	$(c, \bar{c})$	
29	29	$(b, \bar{c})$	
30	30	$(g, \bar{c})$	

as appearing in current fastNLO top grids (note duplications; **fixed in APPLGRID conversions**)

# impact of photon on other pdfs



arXiv:[1712.07053](https://arxiv.org/abs/1712.07053)



# APPLGRID subprocesses for Z production

## APPLGRID subprocesses for $Z^0$ production

We can introduce 12 sub-processes in Z production (calculated using MCFM)

$$U\bar{U} : F^{(0)}(x_1, x_2, Q^2) = U_{12}(x_1, x_2)$$

$$D\bar{D} : F^{(1)}(x_1, x_2, Q^2) = D_{12}(x_1, x_2)$$

$$\bar{U}U : F^{(2)}(x_1, x_2, Q^2) = U_{21}(x_1, x_2)$$

$$\bar{D}D : F^{(3)}(x_1, x_2, Q^2) = D_{21}(x_1, x_2)$$

$$gU : F^{(4)}(x_1, x_2, Q^2) = G_1(x_1)U_2(x_2)$$

$$g\bar{U} : F^{(5)}(x_1, x_2, Q^2) = G_1(x_1)\bar{U}_2(x_2)$$

$$gD : F^{(6)}(x_1, x_2, Q^2) = G_1(x_1)D_2(x_2)$$

$$g\bar{D} : F^{(7)}(x_1, x_2, Q^2) = G_1(x_1)\bar{D}_2(x_2)$$

$$Ug : F^{(8)}(x_1, x_2, Q^2) = U_1(x_1)G_2(x_2)$$

$$\bar{U}g : F^{(9)}(x_1, x_2, Q^2) = \bar{U}_1(x_1)G_2(x_2)$$

$$Dg : F^{(10)}(x_1, x_2, Q^2) = D_1(x_1)G_2(x_2)$$

$$\bar{D}g : F^{(11)}(x_1, x_2, Q^2) = \bar{D}_1(x_1)G_2(x_2)$$

We separate  $u\bar{u}$  from  $\bar{u}u$   
contributions to include  
 $\gamma/Z$  interference

# APPLGRID subprocesses for Z production

## APPLGRID subprocesses for $Z^0$ production II

Use is made of the generalised PDFs defined as:

$$U_H(x) = \sum_{i=2,4,6} f_{i/H}(x, Q^2), \quad \bar{U}_H(x) = \sum_{i=2,4,6} f_{-i/H}(x, Q^2),$$

$$D_H(x) = \sum_{i=1,3,5} f_{i/H}(x, Q^2), \quad \bar{D}_H(x) = \sum_{i=1,3,5} f_{-i/H}(x, Q^2),$$

$$U_{12}(x_1, x_2) = \sum_{i=2,4,6} f_{i/H_1}(x_1, Q^2) f_{-i/H_2}(x_2, Q^2),$$

$$D_{12}(x_1, x_2) = \sum_{i=1,3,5} f_{i/H_1}(x_1, Q^2) f_{-i/H_2}(x_2, Q^2),$$

$$U_{21}(x_1, x_2) = \sum_{i=2,4,6} f_{-i/H_1}(x_1, Q^2) f_{i/H_2}(x_2, Q^2),$$

$$D_{21}(x_1, x_2) = \sum_{i=1,3,5} f_{-i/H_1}(x_1, Q^2) f_{i/H_2}(x_2, Q^2),$$